Abusing Bleeding Edge Web Standards for AppSec Glory

Bryant Zadegan
Advisor/Mentor
Mach37

keybase.io/bryant
@eganist

Ryan Lester
CEO, Co-Founder
Cyph

hacker@linux.com
@TheRyanLester

@eganist

- Does AppSec stuff, usually.
- Mentors security startups, sometimes.
- “Mentors” others on AppSec, occasionally.
- Paid a buck to make Steve Ballmer dance, but just once.

@TheRyanLester

- Runs an E2EE communication startup
- Codes for an E2EE communication startup
- Ran QA automation at a rocket factory
- Got sued by Napster (and not for piracy)

Bleeding Edge Web Standards

- For Your (Ab)use
  - **SRI Fallback**
  - **CSP Meta-Hardening**
  - **HPKP Suicide**

Potential Pain:

Pinch

Punch

Gallstone

But Why?

- New standards are frequently drafted.
- Many introduce unforeseen complications.
- Novel uses encourage future tweaks.

Source: Harold & Kumar Go to White Castle

SubResource Integrity

- Validate resources beyond your trust (e.g. CDNs)
```
<script
 src="https://code.jquery.com/jquery.min.js"
 integrity="sha256-[hash] sha256-[hash2]"
 crossorigin="anonymous">
</script>
```
- [w3.org/TR/SRI/]()
- [caniuse.com/subresource-integrity]()

BUILDER **DEMO**

heisenberg.co/sridemo/

Praise be to the demo gods

CVE-2016-1636 **Demo**

heisenberg.co/sridemo/sameorigin

(͡° ͜ʖ ͡°)

SRI Fallback

Per the SRI Spec:

> NOTE
>
> On a failed integrity check, an `error` event is fired. Developers wishing to provide a canonical fallback resource (e.g., a resource not served from a CDN, perhaps from a secondary, trusted, but slower source) can catch this `error` event and provide an appropriate handler to replace the failed resource with a different one.

...so we implemented it for you.

BUILDER **DEMO**

heisenberg.co/srifallbackdemo/

Kneel to the demo gods

SOURCE (Simplified BSD)

github.com/cyph/sri-fallback

Do source gods even exist?

CSP Meta-Hardening™

- Combines semi-strict header with strict meta.
- Allows for pre-loading of trusted complex logic.
- Does not work for the verbs `frame-ancestors`, `report-uri`, or `sandbox`.

(We didn’t actually trademark this, but it’s a good name.)

BUILDER DEMO

heisenberg.co/metacspdemo/

Fall on thy sword for the demo gods.

CSP Meta-Hardening™

- Best for adapting a semi-recent application for use with CSP.
- Application’s trusted static logic is allowed to execute on initial load.
- Meta-Hardening prevents dynamic content from potentially executing later on.

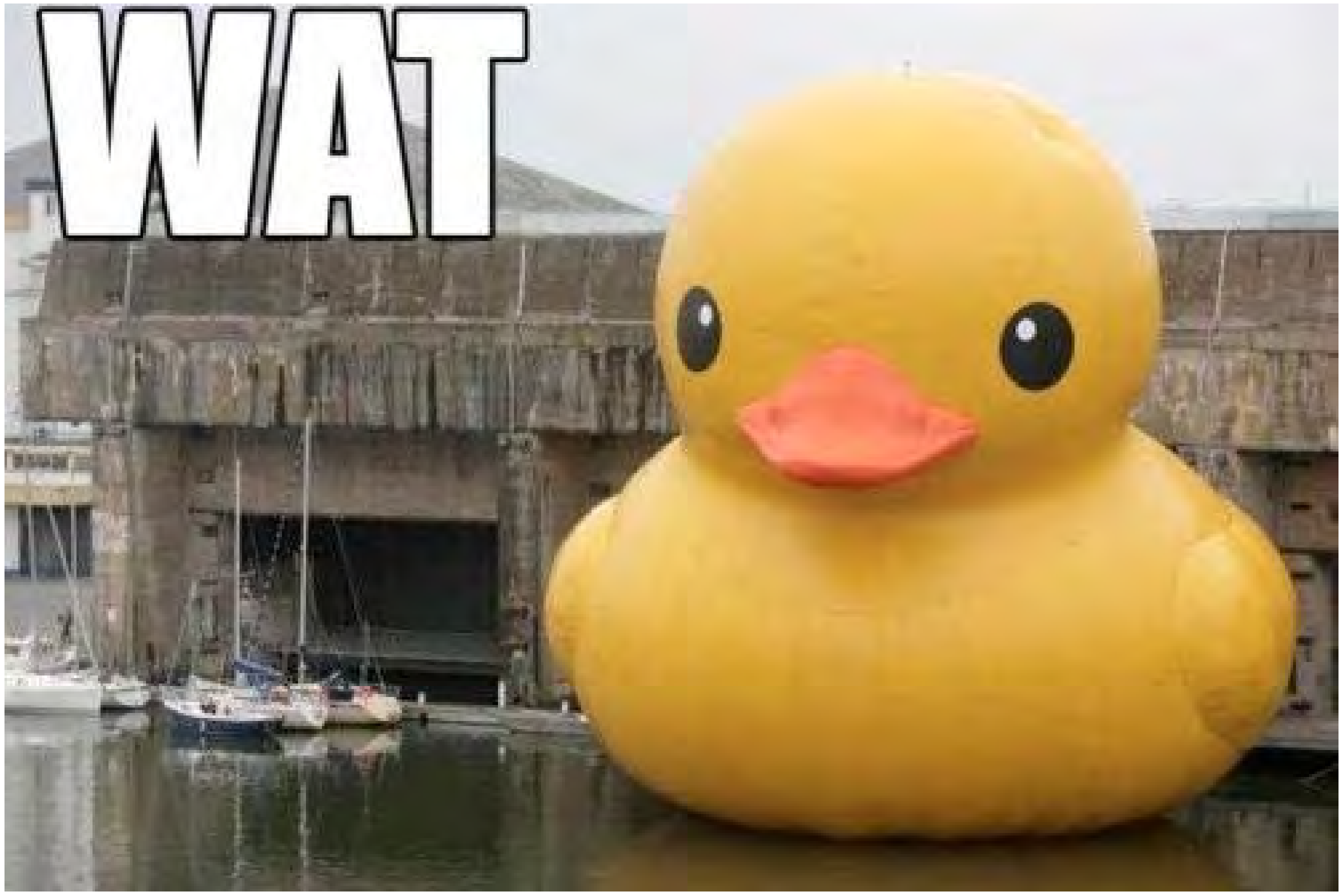
Http Public Key Pinning

- This can ~~break~~ brick sites. Use Reporting!
  - (Chrome 46+ only; no reporting in Firefox 😐)
- Quickstart:

```
Public-Key-Pins-Report-Only:
max-age=5184000; includeSubdomains;
pin-sha256="az9AwClWuHM+fYV+d8Cv9B4sAwdcoUqj93omk18O/pc=";
pin-sha256="5UONcYAsFtYscIlFlm4+aodoL20RRHzGaOeoSNEZ+iA="
report-uri="https://report-uri.io/report/[id]/reportOnly"
```

- tools.ietf.org/html/rfc7469
- caniuse.com/hpkp

HPKP Suicide™



HPKP Suicide™

HPKP + Rapid Key Rotation can trap content:

- ~~to enable in-browser code signing~~
- control content changes and harden SRI.
- to enable nuanced web content blocking. (NetSec)
- to track users...
- to be total jerks...

...in ways we shouldn't put in print.

(Thanks Jann Horn @ Cure53 for putting us onto this!)

HPKP Suicide™ for Builders

Wait, in-browser code signing? No extensions?

- Used HPKP Suicide to pin code-signing logic and keys into the AppCache/Service Worker.
- Logic fetches and validates content from a different origin. Nearly Trust-On-First-Use.

This was so novel, Cyph had to file for a patent (protecting this is why this slide is even here), but you come fairly close to this *for free* if you...

HPKP Suicide™ for Builders

Control local storage updates! Harden SRI!

- Set HPKP max-age to around your deployment schedule, but no more than 60 days.
- Rotate routinely.

Benefits: retain control of front-end content between releases, mitigate risks of SRI hash tampering server-side.

BUILDER **DEMO**

redskins.io

I don't believe in demo gods

HPKP Suicide™
for Builders

Web Content Gateway e.g. [SomeVendor]?
Lock your users out of sites even when they're not on your network!

1. For flagged domains, set HPKP headers.
2. Rotate keys weekly at the web gateway.

Done!

(By us disclosing it, is this now prior art? ☺)

HPKP Suicide™
for Builders

Oh… https://crt.sh/?id=19538258

```
Issuer:
commonName             = VeriSign Class 3 Public
                         Primary Certification
                         Authority - G5

Subject:
commonName             = Blue Coat Public Services
                         Intermediate CA
organizationalUnitName = Symantec Trust Network
organizationName       = "Blue Coat Systems, Inc."
```

HPKP Suicide™ for Builders

User tracking?

Well, we really shouldn't talk about this...

HPKP Suicide™

for **Builders**

But since this is DEF CON…

…let's track users!

HPKP Suicide™

for Builders

Pre-requisites:

1. Lots of (sub)domains to pin
2. Browsers that allow HPKP incognito
3. Rapid Key Rotation

Let's Encrypt

(Thanks! ☺)

HPKP Suicide™

HPKP SuperCookies

Server setup:

1. Point `*.cyph.wang` at the backend server

2. Set `POST /set` to add `${clientIP}-${subdomain}` to cache and return 200 response with valid HPKP header

3. Set `GET /check` to return 418 error response if `${clientIP}-${subdomain}` is in the cache; otherwise return 200 response (no HPKP header)

4. Set a 12-hour interval to delete the current TLS key + IP cache then generate a new TLS key + cert for `[0-31].cyph.wang` and `$(date +%s).cyph.wang`

HPKP Suicide™

HPKP SuperCookies

Client JavaScript:

1. GET `[0-31].cyph.wang/check` and reconstruct a uint32 ID from the resulting binary (with each successful request being 0 and each failure being 1)

2. If ID is 4294967295 (max uint32), assume an error and throw an exception

3. If ID is 0, generate a new ID via `crypto.getRandomValues`, convert it to binary, iterate over the bits, and POST each 1 bit to the correct index within `[0-31].cyph.wang/set`

4. Return final ID to calling code

HPKP Suicide™

HPKP SuperCookies

Considerations:

Risk: DoSing tracker domains as a public service

1. Domain whitelist for your own tracker, or
2. App-issued and tracker-verified nonce if analytics is your business model.

The pattern we described is among those here:

https://tools.ietf.org/html/rfc7469#section-5

BUILDER **DEMO**

cyph.wang

I don't believe in demo gods

SOURCE (New BSD)

github.com/cyph/hpkp-supercookie

Do source gods even exist?

HPKP Suicide™ for Builders

…to be total jerks?

we ***really*** shouldn't talk about this…

HPKP Suicide™
for Breakers

Who are we kidding?

This is DEF CON.

HPKP Suicide™ for Breakers

Pre-requisites:

1. A high-traffic target
2. A way to shell the box
3. A free certificate authority

Let's Encrypt

(Sorry ☹)

HPKP Suicide™

RansomPKP

1. Determine target[1]
2. Generate ransom keypair (the recovery key)
3. Pwn[2] target webserver.
4. Generate new lockout keypair + CSR[3]
5.
6. Profit!

HPKP Suicide™

RansomPKP

While owned users < *n*

1. "public-key-pins:
   max-age=31536000; includeSubdomains;
   pin-sha256= LOCKOUT_KEY;
   pin-sha256= RANSOM_KEY"
2. If owned users = *n*,
   1. Generate new lockout keypair + CSR[3]
   2. Blow old lockout keypair. This locks out *n* users.
   3. $n = 0$

Breaker Demo

isis.io

We're going to regret this.

HPKP Suicide™

Ransom**PKP**

Considerations (i.e. why this is *not* a High):

1. Let's Encrypt limits you to 20 certs per week.
2. Chrome + Firefox have HPKP lockout mitigations (more later)
3. You still need to pop the box.

HPKP Suicide™

Ransom**PKP**

Programmatic Mitigations:

Chromium:
WontFix

Firefox:
Resolution Pending

Let's Encrypt:
WontFix

HPKP Suicide™

RansomPKP

Host Mitigations

1. Use DNS Certification Authority Authorization (CAA) – RFC 6844.
2. Use HPKP.
3. Don't get popped.

HPKP Suicide™

RansomPKP

End User Mitigations (Clearing key pins):

1. Chrome: chrome://net-internals/#hsts
2. ~~Chrome: (alt): clear any irrelevant part of your cache. "due to a curly brace mishap, we've been clearing it over-aggressively for years."~~
   (yes, we reported this one too. CVE-2016-1694)
3. Firefox: about:config >> security.cert_pinning.enforcement_level = 0, visit site to take new header, re-enable.

SOURCE (New BSD)

github.com/cyph/ransompkp

Do source gods even exist?

Hat Tip

To Geller Bedoya, Jonn Callahan, Jann Horn (and all of Cure53), Samy Kamkar, Jim Manico, Mike McBryde, Garrett Robinson, and John Wilander, as well as the Chrome, Firefox, and Let's Encrypt security teams for their contributions.

Thank You!

Bryant Zadegan

Advisor/Mentor

Mach37

keybase.io/bryant

@eganist

Ryan Lester

CEO, Co-Founder

Cyph

hacker@linux.com

@TheRyanLester